REMARKS/ARGUMENTS

These remarks are made in response to the Final Office Action of January 4, 2006 (Office Action). As this response is timely filed within the 3-month shortened statutory period, no fee is believed due.

Claims 1-5, 6, and 9-18 were rejected at page 2 of the Office Action under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,088,671 to Gould, *et al.* (hereinafter Gould). At page 5 of the Office Action, Claims 8 and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Gould in view of U.S. Patent No. 6,539,080 to Bruce, *et al.* (hereinafter Bruce). Claims 7 and 19 were also rejected at page 5 under 35 U.S.C. § 103(a) as being unpatentable over Gould.

Applicants have amended each of independent Claims 1, 9, and 13 to further emphasize certain aspects of Applicants' invention and have cancelled both dependent Claims 2 and 10. Applicants also have amended dependent Claims 3 and 15 to maintain their consistency with the claims from which they each depend. The claim amendments, as discussed herein, are fully supported throughout the Specification. (See, e.g., Specification, p. 3, lines 8-14; p. 6, lines 12-20; p. 8, line 6 – p. 10, line 12.) No new subject matter has been introduced by virtue of the claim amendments presented herein.

Applicants' Invention

It may be useful at this juncture to reiterate certain aspects of Applicants' invention. One aspect of Applicants' invention is the auditory presentation of database query results through an audio user interface (AUI) such that each choice extracted through performance of a database query operation can be audibly presented immediately upon its extraction, rather than according to conventional processes by which all matches are extracted and, only then, presented in batch. Applicants' invention permits a user to respond to each choice as soon as that choice is presented, thereby interrupting the

database query operation at any point and foregoing further subsequent presentations of additional choices through the AUI.

One embodiment of the invention, typified by independent Claim 1, as amended, is a method for presenting database query results through an AUI. The method can include initiating a database query operation, which can perform a retrieval of a plurality of database query result items from at least one database. The method additionally can include audibly presenting the retrieved database query result items through the AUI as each query result item is found in at least one database, the query result items being presented sequentially in a one-by-one manner. The presenting step and database query operation can be performed concurrently and can continue either until a particular one of the retrieved database query result items is selected by a user, or until each of the retrieved database query result items has been sequentially presented to the user. The concurrently performed database query operation and audible presentment of retrieved database query result items can terminate when the user supplies through the AUI a speech input that indicates a user selection of a particular one of the retrieved database query result items.

Yet another embodiment of the invention, typified by amended Claim 9, is a system for presenting database query results through an AUI. The system can include a database manager for managing a database query operation on at least one database, the database query operation producing database query result items. The system also can include a dialog manager for managing the presentation of database query result items through the AUI concurrently with the performing of the database query operation. The database manager and dialog manager can be configured to concurrently perform the database query operation and audible presentments of database query result items such that each query result item is presented sequentially as each is identified through the database query operation. The respective managers, moreover, can be configured so that the database query operation and sequential presentment of retrieved query results items

continues either until a particular one of the retrieved database query result items is selected by a user, or until each of the retrieved database query result items has been sequentially presented to the user. Additionally, the respective managers can be further configured such that the concurrently performed database query operation and audible presentment of retrieved database query result items terminate when the user supplies, through the AUI, a speech input indicating a user selection of a particular one of the retrieved database query result items.

The Claims Define Over The Prior Art

Independent Claims 1, 9, and 13 were each rejected in the Office Action as being anticipated by Gould, as noted above. Applicants respectfully maintain, however, that Gould fails to teach, expressly or inherently, every feature recited in independent Claims 1, 9, and 13, as amended.

Claims 1 and 13

With respect to independent Claims 1 and 13, each of the claims recite the following step:

> *initiating a database query operation, the operation retrieving a plurality of database query result items from at least one database.*

Gould is directed to a system and method for recognizing speech and distinguishing between dictation and commands. (Col. 1, lines 35-36; Abstract.) Specifically, Gould recognizes speech-based signals representing elements "corresponding to text to be recognized and command elements to be executed." (Col. 1, lines 36-39; Abstract.) Upon recognition of each of the speech elements by Gould, "the

recognized elements are acted on in a manner which depends on whether [the elements] are text or commands." (Col. 1, lines 39-41; Abstract.)

Gould's procedures, however, have nothing to do with a database query operation that retrieves a plurality of database query results from one or more databases. Gould's system is explicitly a dictation system for determining whether a user-supplied utterance should be interpreted as dictated text or as a speech command. (See Col. 4, lines 49-67.) For example, were a user to utter the phrase "four score and seven years ago," then assuming perfect speech recognition, Gould's system would present the typed text FOUR SCORE AND SEVEN YEARS AGO because the text does not match a predefined, active command. Conversely, if the user were to say "Bold the previous three words," then assuming this matched an active command in a command list or grammar, the system would not type the words, but instead would bold SEVEN YEARS AGO.

Accordingly, the activities described by Gould are peculiar to speech dictation systems; they emphatically are not database queries. In Gould, a user speaks a phrase, a speech recognizer continues accepting user speech input until an end-of-speech detector determines that the user has stopped speaking, and then the speech recognizer determines the most-likely words based on the user utterance. The converting of utterance to recognized words in Gould is based on (1) the recognizer's acoustic model and (2) the recognizer's language model.

The output of the recognizer in Gould is a text string, but this is typical of dictation systems generally. Gould provides an extra step by delaying output of the string in the form of dictated text until after the system has determined whether the string matches an active command, perhaps one in a simple list of commands or a finite-state grammar that defines permissible commands. If the speech is recognized to be a command, then the command is executed. If not, the recognized words are produced as dictated text.

The operations performed by Gould, however, are not comparable to a database query, which by definition is able to return multiple matches. Gould's actions are carried out against text that is produced by a recognizer, not matches identified during a database query and returned from the database. A database query is fundamentally different than the acoustic input recognition performed by Gould because a database query entails submitting to a database a search string, which has a wholly different structure than an acoustic model, a language model, and a finite-state grammar. A database query based on the search string, moreover, may produce no matches, a single match, or multiple matches. In the case of multiple matches, a user indicates which match is the desired one. Applicants' invention does not involve, nor does it require a speech recognition component that takes audio input, detects the end of the speech, and, based on an acoustic model and either a language model or a finite-state grammar, interprets the audio input in order to output a text string. In Gould, a text string is determined to be either dictated text or a command. By contrast, Applicants' invention utilizes a text string to perform a database query.

Independent Claims 1 and 13, further recite the following step:

> *audibly presenting one-by-one the retrieved database query result items through the AUI as each query result item is found in at least one database.*

As already asserted, Gould does not in fact perform a database query operation. Moreover, Gould does not present to a user database query items retrieved by such an operation. The action that follows Gould's comparison of "speech frames" is explicitly described in a portion cited in the Office Action:

> "[T]he CPU compares speech frames representing the user's speech to speech models in the vocabularies to recognize (step 60) the user's speech.

> The CPU then determines (steps 62 and 64) whether the results represent a command or text. Commands include single words and phrases and sentences that are defined by templates (i.e., restriction rules). The templates define the words that may be said within command sentences and the order in which the words are spoken. The CPU compares (step 62) the recognition results to the possible command words and phrases and to command templates, and if the results match a command word or phrase or a command template (step 64), then the CPU sends (step 65a) the application that called the speech recognition software keystrokes or scripting language that cause the application to execute the command, and if the results do not match a command word or phrase or a command template, the CPU sends (step 65b) the application keystrokes or scripting language that cause the application to type the results as text." (Col. 4, line 51 – Col. 5, line 2.)

As the explicit language reveals, Gould does not present to a user the results of a search query. Instead, Gould sends an application, the particular application depending on whether the speech input is a command or dictation. Moreover, Gould does not present a plurality of database query items in a one-by-one sequential fashion. Rather, as explicitly described in the reference, Gould's recognition of speech elements results in the recognized elements being "acted on in a manner which depends on whether [the recognized elements] represent text or commands." (Col. 1, lines 39-42; see also Abstract.) If the recognized elements comprise a command, then "command keystrokes" are sent to an application; otherwise, if the recognized elements comprise text, then "text keystrokes" are sent to an application. (See FIG. 4.)

Accordingly, the result of Gould's comparison of speech frames results only in one of two alternative actions. Gould's choice of a single action, however, has nothing to do

with the presentment of a plurality of database query items. Certainly, the single action choice effected by Gould's comparison is not even remotely related to the one-by-one, sequential presentment of a plurality of database query items. Gould's action of sending a keystroke to an application has nothing to do with the presentment of a database query items, let alone the sequential presentment of a plurality of database query items.

Accordingly, another critical distinction between Gould and Applicants' invention is that Applicants' invention precludes having to wait until all database query items are retrieved by a database query operation and then having to form a disambiguation prompt (e.g., "Was that Sam Hill in Raleigh or Sam Hill in Austin?"). Applicants' invention audibly presents database matches as they are returned, the presentment occurring concurrently with the database query operation. Thus, Applicants' invention provides a mechanism whereby a spoken indication from a user indicates that a just-presented match is or is not the desired match. Gould provides no such mechanism.

It should also be noted that Gould does not even allude to an audible presentment of any kind of item; Gould's presentments are strictly visual. This is made explicit throughout the reference:

> "[W]hile dictating text, the user may cause the computer to display a command browser 66 by keystroke, mouse selection, or utterance (e.g., speaking the phrase "What Can I Say" 68 into the microphone). The command browser displays possible commands for the application being executed. For example, a word processing application includes single command words, e.g., [Bold] 70 and [Center] 72, command phrases, e.g., [Close Document] 74 and [Cut This Paragraph] 76, and flexible sentence commands, e.g., [<Action><2 to 20><Text Objects>] 78 and [Move <Direction><2 to 20><Text Objects>] 80." (Col. 5, lines 3-14) (emphasis supplied.)

"The partial results allow the user to see how the recognition is proceeding. If the speech recognition is not accurate the user can stop speaking and proceed by speaking more slowly or clearly or the user or a technician can use the partial results information to diagnose speech recognition system errors." (Col. 6, line 47-52.) (emphasis supplied.)

Regarding the presenting step, independent Claims 1 and 13 further specify that this step is performed concurrently with the database query operation:

the presenting step occurs concurrently with the database query operation.

Gould does not teach, expressly or inherently, presenting to a user database query items concurrently with performing a database query operation. In a portion cited at page 7 of the Office Action, Gould describes a procedure for keeping a "count" of stored "speech packets" not yet processed:

"Monitor software 32 keeps a count 34 of the number of speech packets stored but not yet processed. An application 36, for example, a word processor, being executed by the CPU periodically checks for user input by examining the monitor software's count. If the count is zero, then there is no user input. If the count is not zero, then the application calls speech recognizer software 38 and passes a pointer 37 to the address location of the speech packet in buffer 30. The speech recognizer may be called directly by the application or may be called on behalf of the application by a separate program, such as DragonDictate.TM. from Dragon Systems.TM.

> of West Newton, Mass., in response to the application's request for input from the mouse or keyboard." (Col. 3, lines 36-50)

This portion of Gould is cited at page 7 of the Office Action as suggesting the managing of queries that have been entered but not yet processed. Yet this has nothing to do with presenting to a user retrieved database query items concurrently with the performing on a database query operation that retrieves such items. The mere keeping a "count" of items yet to be processed, however, has nothing to do with concurrently performing a database query operation and presenting database query items as they are retrieved.

Elsewhere, in another portion cited in the Office Action, Gould describes an on-going comparison of speech frames and probability ranking of possible command inputs:

> "[T]he [speech] recognizer continues by comparing the successive speech frames to the pre-filtered speech models but not to other speech models (e.g., "Bold"). The possible command list is ranked in the order of highest to lowest probability with the command including the speech model most closely matching the speech frames being of highest probability (best candidate). As the CPU continues to compare successive speech frames to the pre-filtered speech models, the CPU actively re-ranks the command list if the probabilities change. (Col. 7, lines 43-52.)

The ranking of speech frames and ranking the probabilities of possible commands based on speech input, however, is wholly unrelated to concurrently performing a database query operation and presenting database query items as they are retrieved.

Moreover, both independent Claims 1 and 13 further specify that the presenting step continues until one of two events occurs:

> *[the step] continuing until a particular one of the retrieved database query result items is selected by a user, or until each of the retrieved database query result items has been sequentially presented to the user;*
>
> *wherein the concurrently performed database query operation and audible presentment of retrieved database query result items terminate when the user supplies through the AUI a speech input indicating a user selection of a particular one of the retrieved database query result items.*

Gould does not expressly or inherently teach that the presentment of database query items continues until either a particular one of a plurality of retrieved database query result items is selected by a user, or until each of the retrieved database query result items has been sequentially presented to the use. In a portion cited at page 3 of the Office Action, Gould describes an embodiment of a speech recognition system that services as "a meeting scheduler:"

> "For example, the application being executed by the system is a meeting scheduler (FIGS. 8a, 8b, 9a, and 9b). After the system displays partial results 302 "schedule this meeting in room 507" (FIG. 8a), the system determines that the utterance was a command and removes the text from the display screen (FIG. 8b) and executes the command by scheduling 304 the meeting in room 507. Similarly, after the system displays partial results 304 "underline last three words" (FIG. 9a), the system determines that the utterance was a command and removes the text from the display screen

(FIG. 9b) and executes the command by underlining 306 the last three words." (Col. 6, lines 34-46.)

The explicit language of this quoted portion of Gould, describes only that a partial result of an action is visually displayed as text before determining that a command is to be executed. This visual display of text – termed "a partial result" – however, has nothing to do with discontinuing an on-going sequential presentment of a plurality of retrieved database query items in response to a user's selecting one of the presented items.

Gould's "meeting scheduler," as described receives only one input, namely, the utterance of a user, and performance of only one action, based on the utterance. Specifically, Gould operates by determining whether the utterance is a user command, and, if so, a single action is performed accordingly. This, however, is wholly distinct from first initiating a database query that results in the concurrent performance of a database query operation along with the sequential presentment of database query items retrieved by the operation, followed by a termination of the concurrent performance and presentment based on a user selection of one of the items. With Gould, there is no presentment of a plurality of items from which a user selects so as to terminate a concurrently performed operation and presentment. The Gould user simply makes a single utterance, based on which the requisite command is executed.

Claim 9

Independent Claim 9 is directed to a system for presenting database query results through an audio user interface (AUI). The system explicitly includes

> *a database manager for managing a database query operation on at least one database, the database query operation producing database query result items.*

Applicants respectfully maintain that, for the reasons already stated, Gould does not provide such a database manager. Specifically, Gould is directed to a dictation system for determining whether a user-supplied utterance is to be interpreted as dictated text or as a speech command. As already stated, Gould's dictation system has nothing to do with a database query operation that retrieves a plurality of database query results from one or more databases.

Independent Claim 9 further recites that the system for presenting database query results through an audio user interface (AUI) additionally includes

> *a dialog manager for managing the presentation of said database query result items through the AUI concurrently with said database query operation.*

Gould does not expressly or inherently teach a dialog manager that manages an audible presentation of database query results concurrently with database operations. In the portion noted at page 7 of the Office Action, Gould states

> "[i]nterrupt signal 26 also causes the operating system software to call monitor software 32. Monitor software 32 keeps a count 34 of the number of speech packets stored but not yet processed. An application 36, for example, a word processor, being executed by the CPU periodically checks for user input by examining the monitor software's count. If the count is zero, then there is no user input. If the count is not zero, then the application calls speech recognizer software 38 and passes a pointer 37 to the address location of the speech packet in buffer 30. The speech

recognizer may be called directly by the application or may be called on behalf of the application by a separate program, such as DragonDictate.TM. from Dragon Systems.TM. of West Newton, Mass., in response to the application's request for input from the mouse or keyboard." (Col. 3, lines 36-49.)

Gould does not here teach, expressly or inherently, a dialog manager that manages the audible presentation of database query results concurrently with the database operation. Rather Gould is describing the monitoring of speech packets (speech input) that are monitored to determine whether an audio input is to be processed for the purpose of speech recognition. Gould, however, teaches nothing about the audible presentation of database query results.

Independent Claim 9, further specifies the precise configuration of the database manager and the dialog manager of the system for presenting database query results through an audio user interface (AUI):

the database manager and dialog manager are configured to concurrently perform the database query operation and audible presentments of database query result items such that the each query result item is presented sequentially as each is identified through the database query operation, the sequential presentment continuing until a particular one of the retrieved database query result items is selected by a user, or until each of the retrieved database query result items has been sequentially presented to the user, and

the database manager and dialog manager are further configured to terminate the concurrently performed database query operation and audible presentment of retrieved database query result items when the user

supplies through the AUI a speech input indicating a user selection of a particular one of the retrieved database query result items.

For each of the various reasons asserted above, Gould fails to expressly or inherently teach a database manager and dialog manager configured so as to cooperatively perform the recited functions. Specifically, Gould does not expressly or inherently teach a system that concurrently performs a database query operation and audible presentment of database query result items such that the each query result item is presented sequentially as each is identified through the database query operation. Nor does Gould expressly or inherently provide a system in which a sequential presentment continues either until a particular retrieved database query result item is selected by a user, or until each of the retrieved database query result items has been sequentially presented to the user. Likewise Gould does not expressly or inherently teach a system that terminates a concurrently performed database query operation and audible presentment of retrieved database query result items when the user supplies through the AUI a speech input indicating a user selection of a particular database query result item.

Applicants respectfully maintain that Gould further fails to expressly or inherently teach using a text-to-speech processor as recited in dependent Claim 10. In another portion cited at page 7 of the Office Action, Gould states

> "As an alternative to dictating directly to an application, the user dictates text to a speech recognizer window, and after dictating a document, the user transfers the document (manually or automatically) to the application." (Col. 4, lines 12-15.)

This portion of Gould does not explicitly or inherently describe a text-to-speech processor. Instead, it describes the use of a special interface for holding the results of

speech recognition (speech-to-text processing) before committing the recognized speech to an application via manual or automatic transfer. Gould nowhere describes the use of text-to-speech (TTS) processing.

Applicants respectfully maintain that Gould does not expressly or inherently teach a "barge-in facility" as explicitly recited in dependent Claim 11. In the portion of Gould cited at page 7 of the Office Action, Gould states

> "[i]f the CPU determines that the user's speech is text and the partial results match the final results, then the CPU is finished. However, if the CPU determines that the user's speech is text but that the partial results do not match the final results, then the CPU sends keystrokes or scripting language to the application to correct the displayed text. Similarly, if the CPU determines that the user's speech was a command, then the CPU sends keystrokes or scripting language to the application to cause the application to delete the partial results from the screen and execute the command." (Col. 6, lines 24-34.)

Applicants respectfully maintain that Gould is only describing a dictation system that can distinguish between speech input intended to be produced as dictated text and speech input intended for interpretation as a command. It would not be useful to introduce or incorporate a "barge-in facility" into a dictation system. A dictation system must continuously listen for speech input. In Gould, the system is intended to produce some result for any and all speech input: either dictated text or a command. Because the system of Gould does not produce any speech output, there can be nothing into which a user would barge using a barge-in facility. Barge-in only makes sense in conversational systems – that is, systems that employ speech input and speech output – where the barge-

in allows users to interrupt speech output. It follows that Gould can not be read as teaching a barge-in facility.

It is further asserted at page 8 of the Office Action that Bruce teaches that database query results can be presented through an AUI as the results are determined concurrently with the execution of a database operation. In the specific portion quoted at page 8 of the Office Action, Bruce states that

> "In a particular embodiment, the route to the destination location can be mapped taking into account the route traffic, travel-times, road conditions, and route weather conditions. The caller may receive the driving or route instructions in a variety of different ways. The route instructions can be communicated directly over the telephone from an interactive voice response system, a live operator, a synthesized voice, a voice mail message, and Internet electronic mail, an alpha/numeric pager or telephone or a Personal Digital Assistant ('PDA')." (Col. 2, lines 54-63.)

Although Bruce utilizes an AUI, there is not the slightest suggestion in Bruce of presenting the results of the database query as those results are obtained concurrently with the execution of the database operation. Applicants respectfully submit that Bruce is not only silent about the presentation of database matches as they are found, but, in fact, it would be a design mistake. It would be counterproductive to apply such a feature to Bruce because in a system presenting navigation instructions, a user would want instructions to either be available all at once so the user could write them down or, alternatively, to be presented on an as-needed basis driven by a user's requesting the next instruction or by having the application be aware of and responsive to the user's location. It logically follows that Bruce can not be read as inherently teaching the presentment of

database query results through an AUI as the results are determined concurrently with the execution of a database operation.

Fundamental Distinctions

As already noted, the actions described in Gould are those associated with the use of acoustic models, language models, and finite-state grammars. Indeed, Gould describes a comparison of the output of two recognizers: one for dictated text and one for predefined commands. By contrast, Applicants' invention can operate based on standard speech recognition elements, but unlike Gould or conventional systems, Applicants' invention presents matches from a database query that has nothing to do with the speech recognition. Applicants' invention, moreover, enables a user to confirm or disconfirm that a match as been identified through the query operation as soon as the matches are made. This is not expressly or inherently taught by Gould.

The speech processing in Gould is essentially an automatic determination of whether a text string produced by the recognizer should be interpreted as dictated text or a command. Gould presents a user no alternatives during a database query, only a determination as to whether a user input is dictated text or predefined command. Gould provides no mechanism by which a user is able to confirm or disconfirm matches concurrently with the operation of a database query.

The relevant literature supports the contention that the method in Gould is not a database lookup, but rather, speech processing according to operations divorced from the searching of databases. Two highly-regarded books concerning speech recognition/natural language processing are Frederick Jelinek, STATISTICAL METHODS FOR SPEECH RECOGNITION (MIT Press, 1997) and Christopher Manning and Hinrich Schutze, FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING (MIT Press,

1999). The index of the former discloses no entries for "database". The index of the later provides one entry for "database," and one for "database merging" (at p. 530). In the Manning and Schutze text, the topic of database merging is included in the chapter on information retrieval, not in the chapters that describe the mechanics of speech recognition. By contrast, examination of the index of a standard text on VoiceXML programming, C. Sharma and J. Kunins, VoiceXML (Wiley Computer Publishing, 2002), reveals four independent entries for the term "database," including a lengthy description of how to work with databases in a case study of building an application.

The standard literature, accordingly, confirms that the usual programmer's conception of a database is that it is data structured into fields. A telephone directory is a database, typically including fields for a person's name and telephone number. Applications work with databases by adding new data, deleting old data, and looking up data. To look up data, one must submit a query which contains information about part of the record (name) in an attempt to retrieve some other part of the record (telephone number).

Speech recognition as in Gould operates in a manner wholly distinct from database query operations, as performed according to Applicants' invention. For example, database lookups are completely deterministic, whereas speech recognition processes as in Gould are inevitably probabilistic and usually based on hidden Markov models and/or other probabilistic models. In the Jelinek text referenced above, for example, the preface states that "[t]he text concentrates on those basic statistical ideas that have proven so fruitful in speech recognition: hidden Markov models, data clustering, smoothing of probability distributions, the decision tree method of equivalence classification, the use of information measures as goodness criteria, and maximum entropy probability estimation." The relevant literature and common usage thus confirm that Gould does not expressly or inherently teach a database query operation as recited in independent Claims 1, 9, and 13.

Accordingly, Gould does not expressly or inherently teach every feature of independent Claims 1, 9, or 13. Applicants respectfully assert, therefore, that each of the independent claims, as amended, defines over the prior art. Applicants further respectfully assert that Gould in combination with Bruce similarly fails to teach or suggest every feature recited in the claims. Applicants respectfully assert also that whereas each of the remaining dependent claims depends from one of the amended independent claims while reciting additional features, these claims likewise define over the prior art.

CONCLUSION

The Applicants believe that this application is now in full condition for allowance, which action is respectfully requested. The Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the application to completion.

Respectfully submitted,

Date: **March 6, 2006**

Gregory A. Nelson, Registration No. 30,577
Richard A. Hinson, Registration No. 47,652
Marc A. Boillot, Registration No. 56,164
AKERMAN SENTERFITT
Customer No. 40987
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 653-5000